Inventor(s): Curtis EUBANKS
Invention: Method and Apparatus ... Medium
Serial No.: 09/661,878

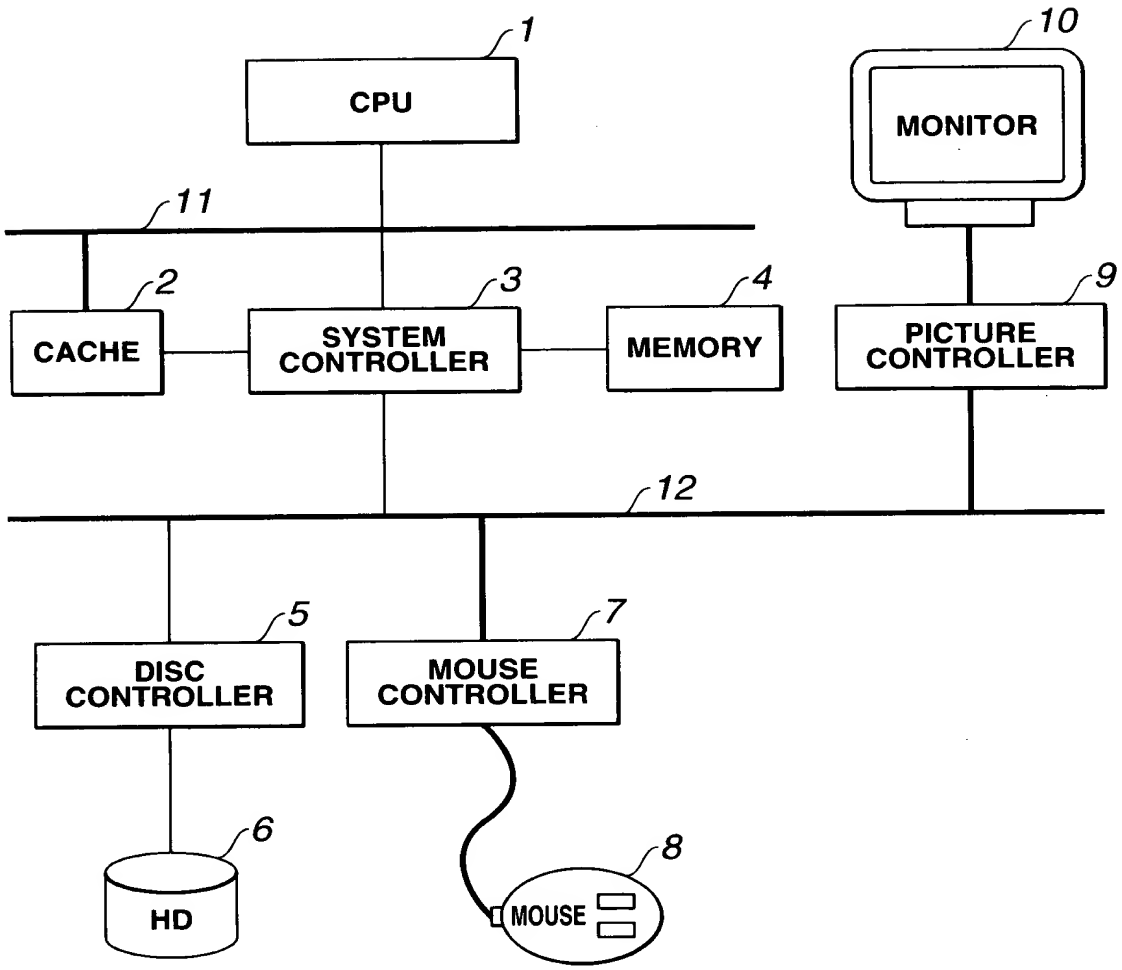

FIG.1

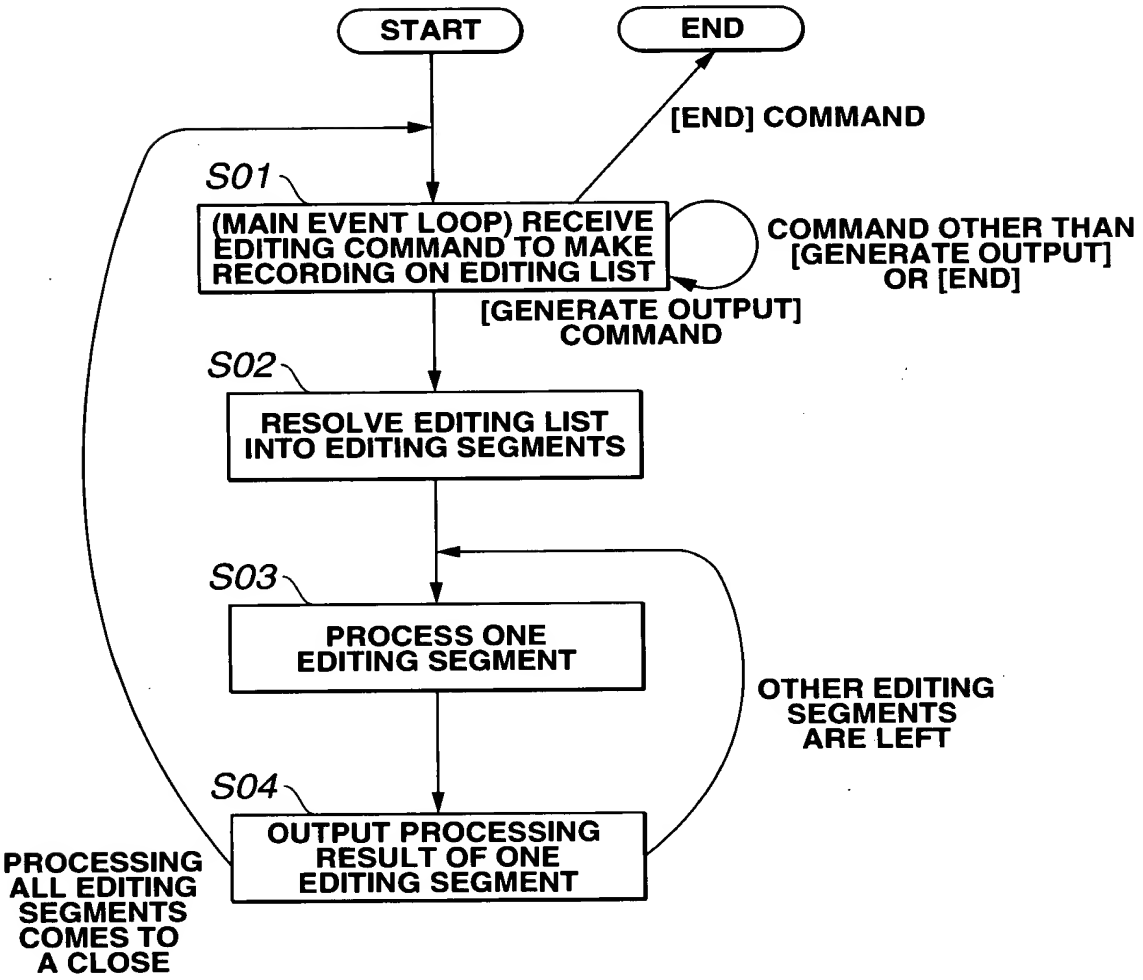
START
END
[END] COMMAND
S01
(MAIN EVENT LOOP) RECEIVE EDITING COMMAND TO MAKE RECORDING ON EDITING LIST
COMMAND OTHER THAN [GENERATE OUTPUT] OR [END]
[GENERATE OUTPUT] COMMAND
S02
RESOLVE EDITING LIST INTO EDITING SEGMENTS
S03
PROCESS ONE EDITING SEGMENT
OTHER EDITING SEGMENTS ARE LEFT
S04
OUTPUT PROCESSING RESULT OF ONE EDITING SEGMENT
PROCESSING ALL EDITING SEGMENTS COMES TO A CLOSE

FIG.2

FIG.3A

```
class EditList {
    int     startTime;     /*  START TIME (ABSOLUTE FRAME)                */
    int     duration;      /*  ENTIRE VIDEO TIME(FRAME)                   */
    int     aClipCounty;   /*  NUMBER OF CLIPS IN TRACK A                 */
    AVClip  aSources[];    /*  CLIP DATA OF EACH CLIP IN TRACK A          */
    int     bClipCount;    /*  NUMBER OF CLIPS IN TRACK A                 */
    AVClip  bSources[];    /*  CLIP DATA OF EACH CLIP IN TRACK B          */
    int     effectsCount;  /*  NUMBER OF EFFECTS IN EFFECT TRACK          */
    Effect  effects[];     /*  EFFECT DATA                                */
};
```

FIG.3B

```
class AVClip {
    int     startTime;     /*  START TIME (ABSOLUTE FRAME)                */
    int     duration;      /*  NUMBER OF FRAMES IN THIS CLIP              */
    int     mediaType;     /*  VIDEO COMPRESSION FORMAT (STANDARD)        */
    char    filename[];    /*  FILENAME OF THIS INPUT STREAM              */
    int     frameOffset;   /*  FILE START FRAME                           */
};
```

FIG.3C

```
class Effect {
    int     startTime;     /*  START TIME (ABSOLUTE FRAME)                */
    int     duration;      /*  NUMBER OF FRAMES IN THIS EFFECT            */
    int     effectType;    /*  STANDARD FORMAT EFFECT TYPE (E.G. WIPE)    */
    void    *effectParams; /*  VARIABLE EFFECT PARAMETERS (START %,       */
                           /*  END OP; E.G. DIRECTION)                    */
};
```

FIG.4A
FIG.4B
FIG.4C
FIG.4D

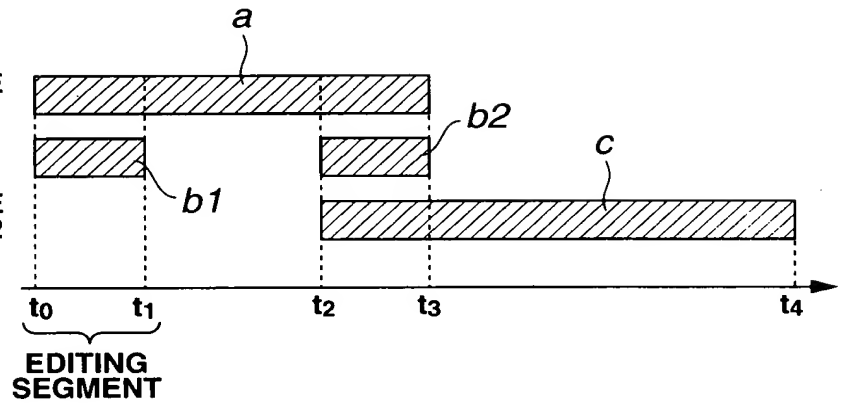

FIG.5A

EDITING SEGMENTS 1	
START	t_0
PERIOD	$t_1 \sim t_0$
VIDEO 1	a
VIDEO 2	⊘
EFFECT	b1

FIG.5B

EDITING SEGMENTS 2	
START	t_1
PERIOD	$t_2 \sim t_1$
VIDEO 1	a
VIDEO 2	⊘
EFFECT	⊘

FIG.5C

EDITING SEGMENTS 3	
START	t_2
PERIOD	$t_3 \sim t_2$
VIDEO 1	a
VIDEO 2	c
EFFECT	b2

FIG.5D

EDITING SEGMENTS 4	
START	t_3
PERIOD	$t_4 \sim t_3$
VIDEO 1	⊘
VIDEO 2	c
EFFECT	⊘

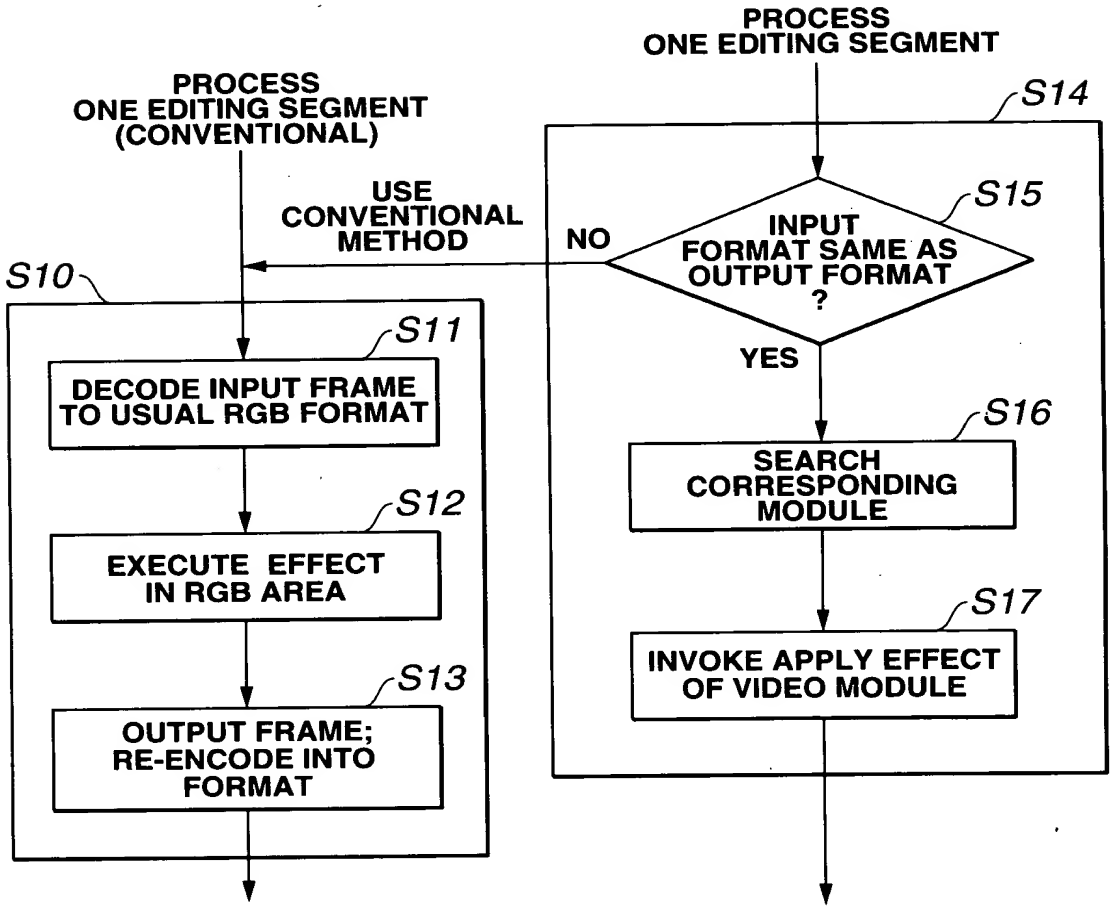
PROCESS ONE EDITING SEGMENT (CONVENTIONAL)
PROCESS ONE EDITING SEGMENT
S14
USE CONVENTIONAL METHOD
NO
S15
INPUT FORMAT SAME AS OUTPUT FORMAT ?
YES
S10
S11
DECODE INPUT FRAME TO USUAL RGB FORMAT
S16
SEARCH CORRESPONDING MODULE
S12
EXECUTE EFFECT IN RGB AREA
S17
INVOKE APPLY EFFECT OF VIDEO MODULE
S13
OUTPUT FRAME; RE-ENCODE INTO FORMAT

FIG.6

```
class VideoModel {
  int format;     /*  STANDARD FORMAT IDENTIFIER   */
  int class;      /*  STANDARD CLASS IDENTIFIER  */

  // DECODE FRAME INTO RGB
  char *ConventToBaseband(AVSource *pSource, int frame);

  // DECODE FRAME TO FORMAT (TO E.G. DCT LEVEL)
  void *ConvertToClass(AVSource* pSource);

  // TRANSFORM FROM FORMAT CLASS(E.G. DCT) TO THIS FORMAT
  void *ConvertFromClass(Void* pClassData);

  // APPLY TRANSITION EFFECT USING SPECIFIED KNOWLEDGE OF THIS FORMAT
  int ApplyEffect(AVSource *pInputA, AVSource *pInputB,
                  AVSource *pOutputC, Effect *pEffect);
};
```

FIG.7

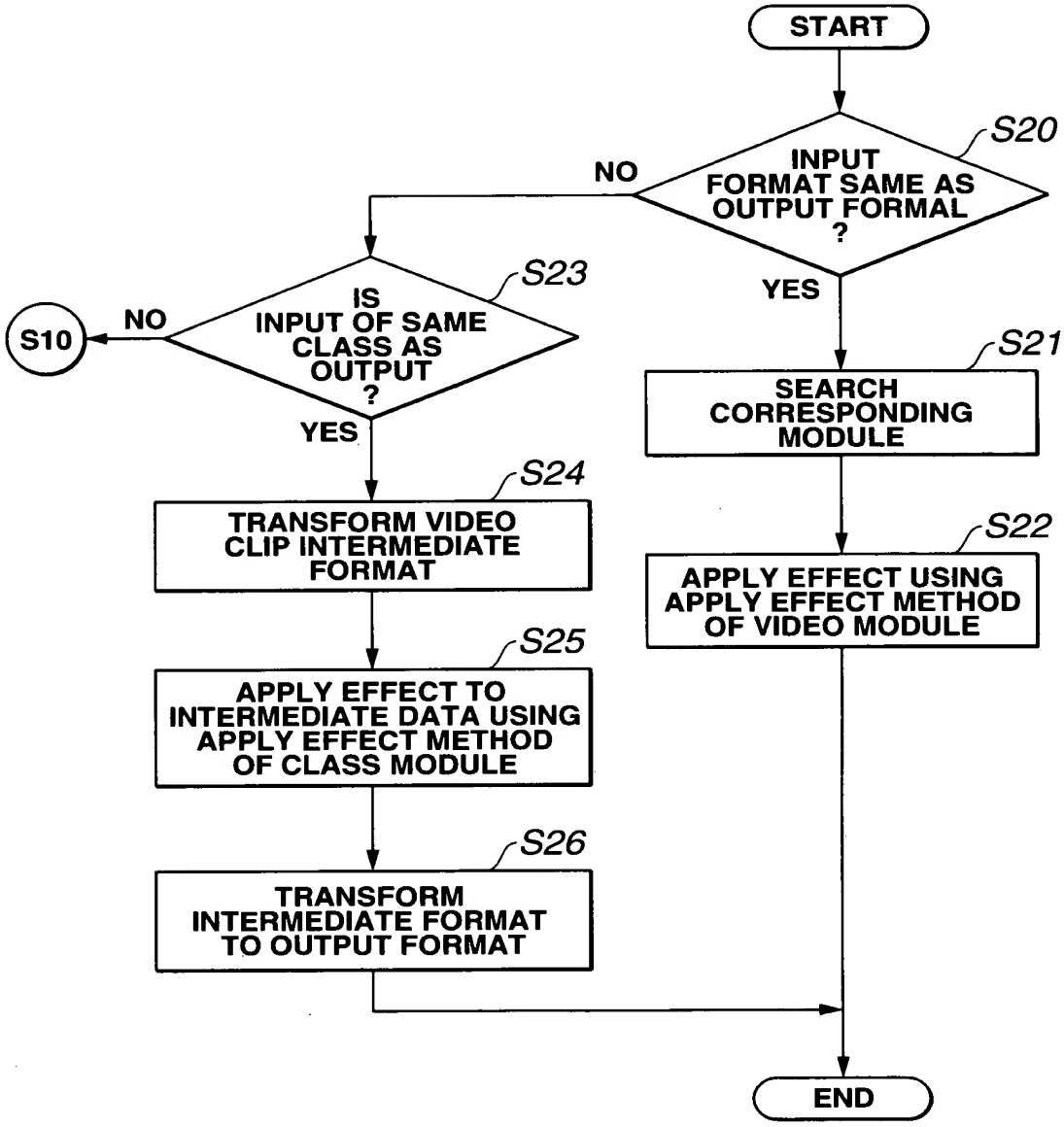
START
S20
INPUT FORMAT SAME AS OUTPUT FORMAL ?
NO
YES
S23
IS INPUT OF SAME CLASS AS OUTPUT ?
NO
S10
YES
S21
SEARCH CORRESPONDING MODULE
S24
TRANSFORM VIDEO CLIP INTERMEDIATE FORMAT
S22
APPLY EFFECT USING APPLY EFFECT METHOD OF VIDEO MODULE
S25
APPLY EFFECT TO INTERMEDIATE DATA USING APPLY EFFECT METHOD OF CLASS MODULE
S26
TRANSFORM INTERMEDIATE FORMAT TO OUTPUT FORMAT
END

FIG.8

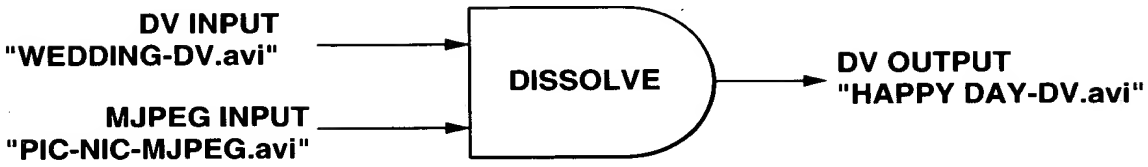

FIG.9

```
#define FourCC(a,b,c,d)      ((a<<24) | (b<<16) | (c<<8) | (d))

#define FORMAT/DV            FourCC('D','V','C','S')
#define FORMAT_MJPG          FourCC('M','J','P','G')

#define CLASS_IDCT           FourCC('I','D','C','T')
```

FIG.10

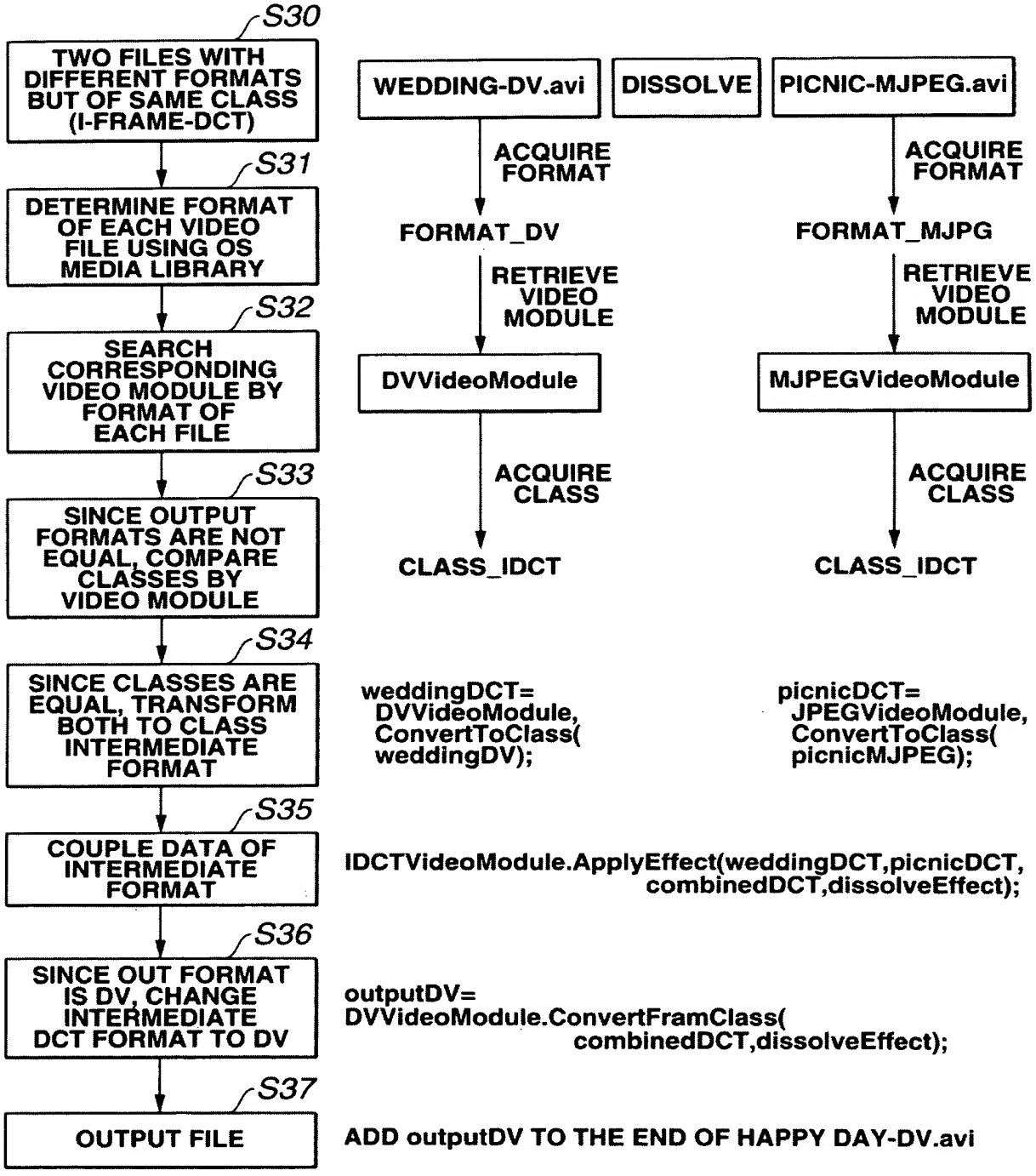
S30
TWO FILES WITH DIFFERENT FORMATS BUT OF SAME CLASS (I-FRAME-DCT)
S31
DETERMINE FORMAT OF EACH VIDEO FILE USING OS MEDIA LIBRARY
S32
SEARCH CORRESPONDING VIDEO MODULE BY FORMAT OF EACH FILE
S33
SINCE OUTPUT FORMATS ARE NOT EQUAL, COMPARE CLASSES BY VIDEO MODULE
S34
SINCE CLASSES ARE EQUAL, TRANSFORM BOTH TO CLASS INTERMEDIATE FORMAT
S35
COUPLE DATA OF INTERMEDIATE FORMAT
S36
SINCE OUT FORMAT IS DV, CHANGE INTERMEDIATE DCT FORMAT TO DV
S37
OUTPUT FILE
WEDDING-DV.avi
DISSOLVE
PICNIC-MJPEG.avi
ACQUIRE FORMAT
ACQUIRE FORMAT
FORMAT_DV
FORMAT_MJPG
RETRIEVE VIDEO MODULE
RETRIEVE VIDEO MODULE
DVVideoModule
MJPEGVideoModule
ACQUIRE CLASS
ACQUIRE CLASS
CLASS_IDCT
CLASS_IDCT
weddingDCT= DVVideoModule, ConvertToClass(weddingDV);
picnicDCT= JPEGVideoModule, ConvertToClass(picnicMJPEG);
IDCTVideoModule.ApplyEffect(weddingDCT,picnicDCT, combinedDCT,dissolveEffect);
outputDV= DVVideoModule.ConvertFramClass(combinedDCT,dissolveEffect);
ADD outputDV TO THE END OF HAPPY DAY-DV.avi

FIG.11

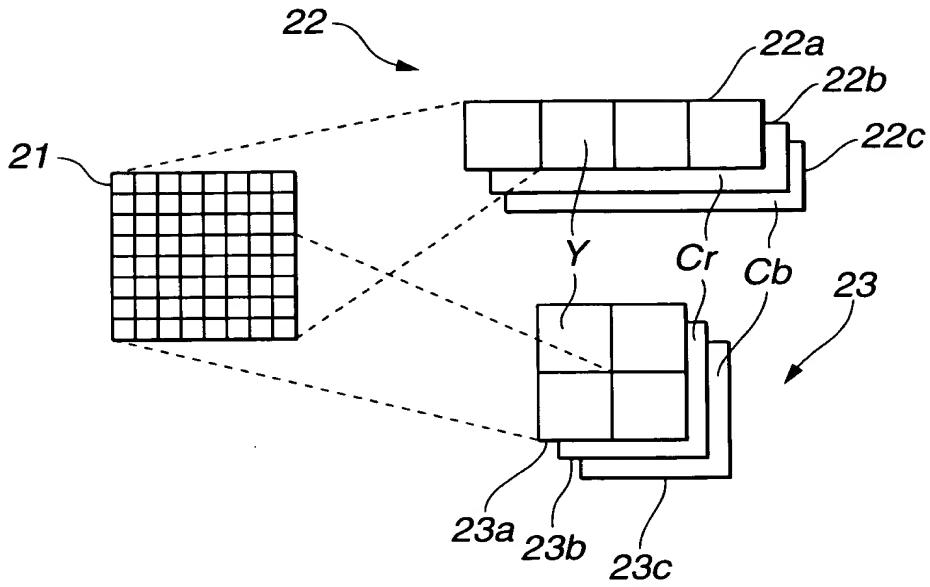
22
22a
22b
22c
21
Y
Cr
Cb
23
23a
23b
23c

FIG.12

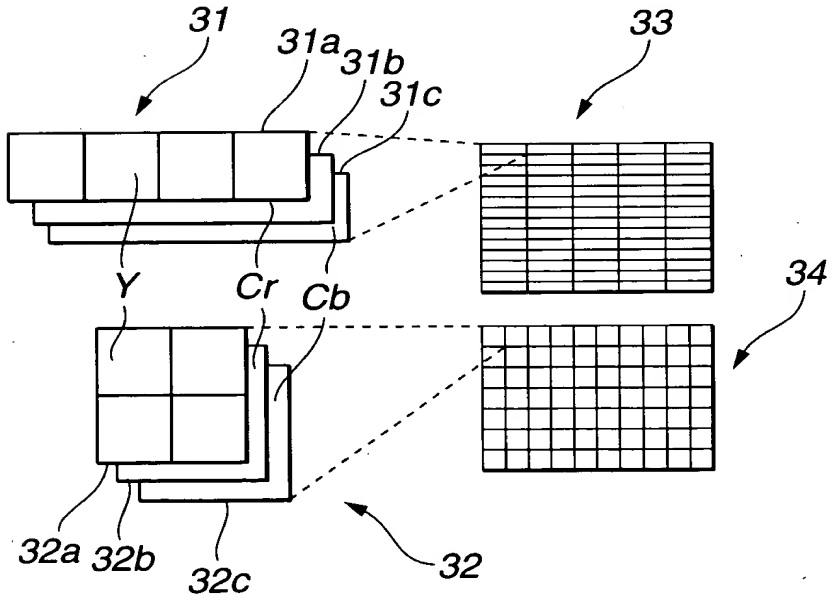
31
31a
31b
31c
33
Y
Cr
Cb
34
32a
32b
32c
32

FIG.13

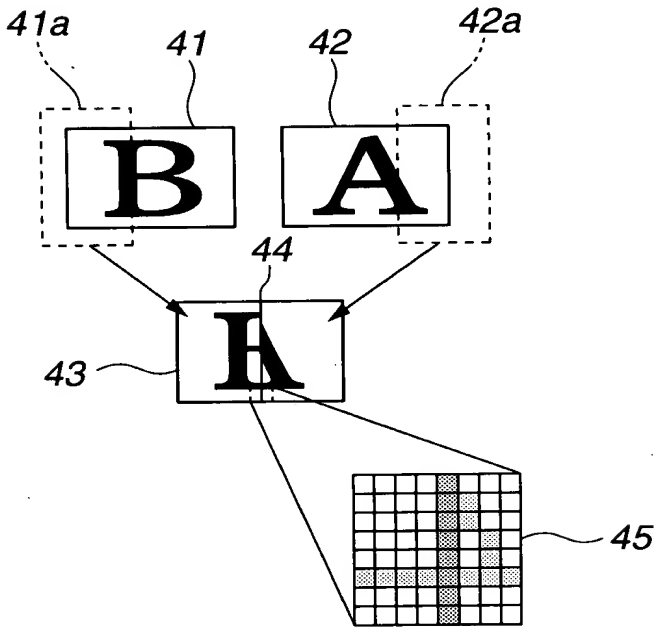

FIG.14